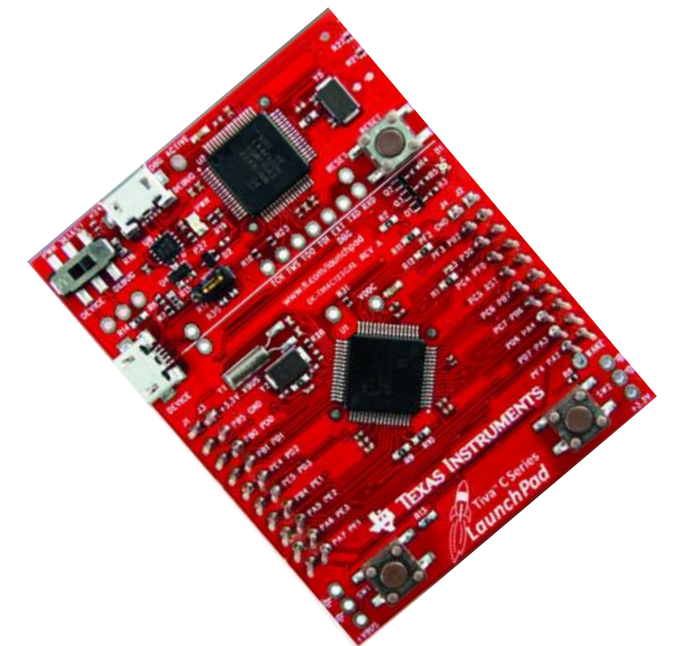
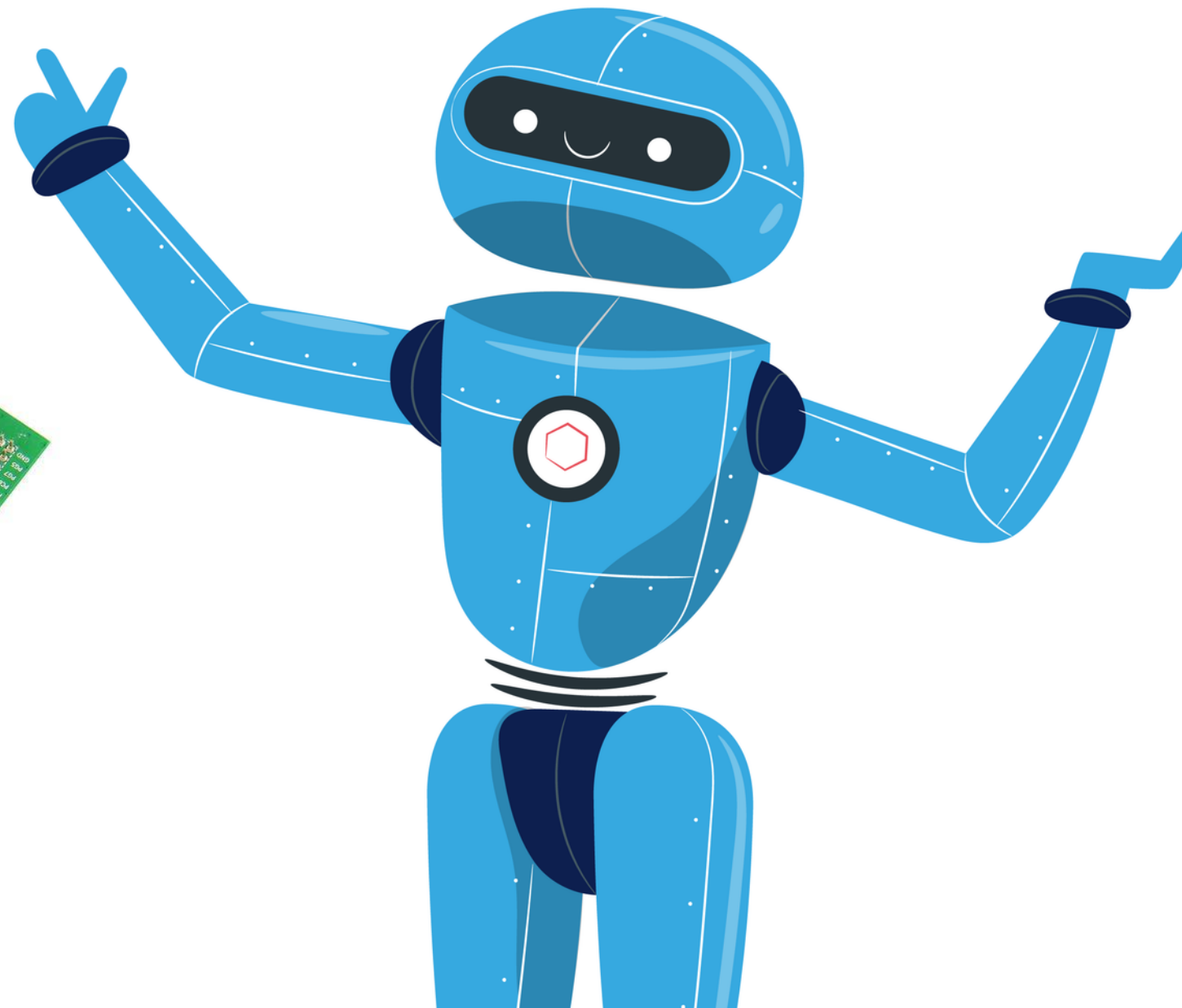
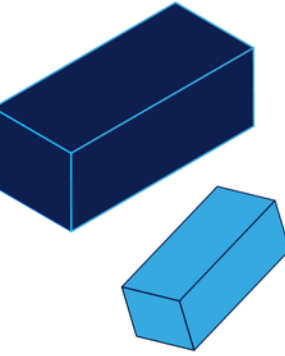


CLASE 7: UART

MICROCONTROLADORES ARM





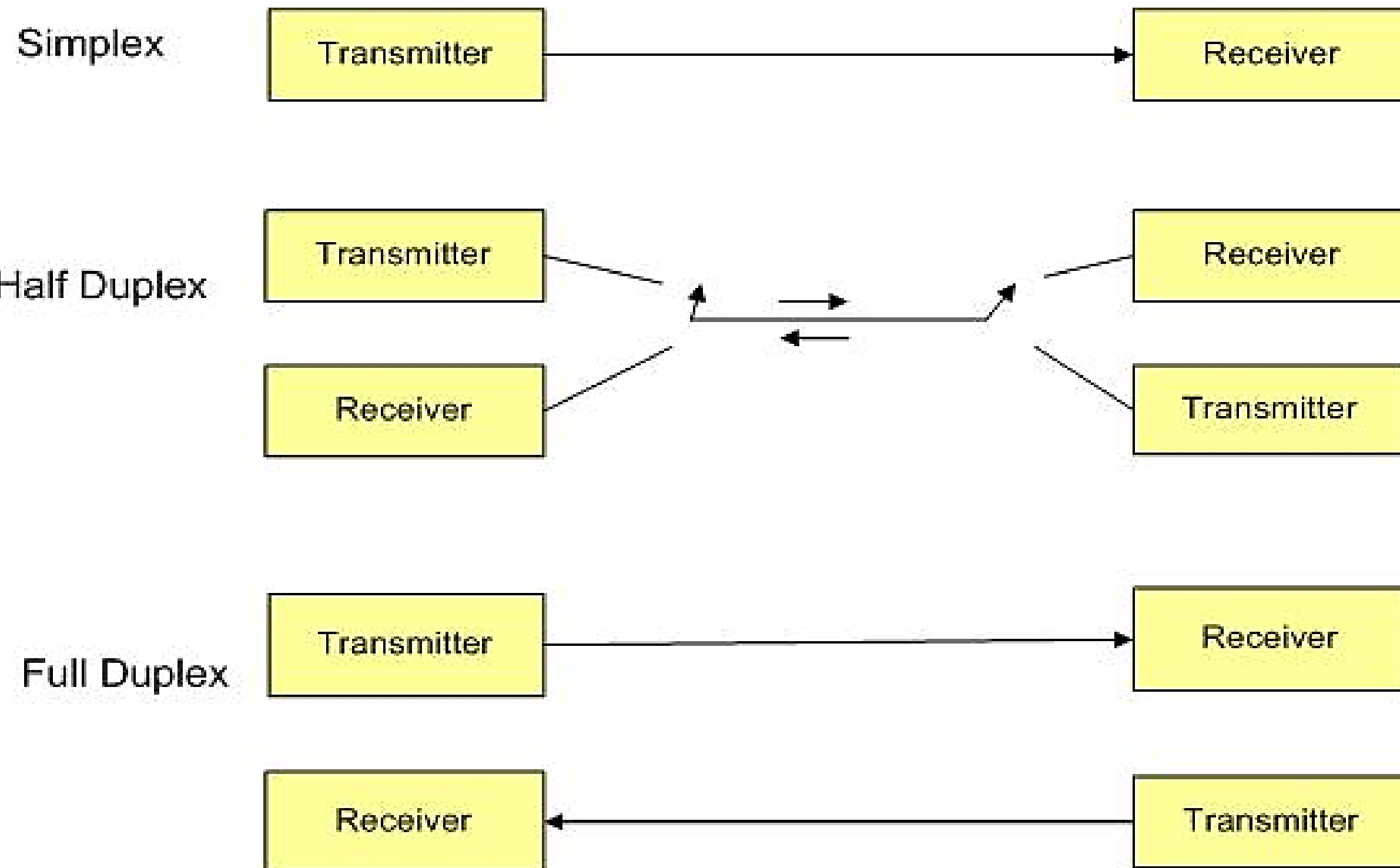
UART

arm

MICRO-
CONTRO-
LADORES
ARM

UMAKER | CENTRO DE CAPACITACIÓN
DE DESARROLLO TECNOLÓGICO

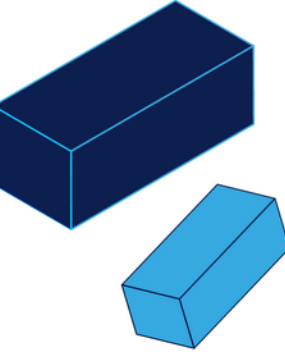
Simplex, Half-, and Full-Duplex Transfers



arm

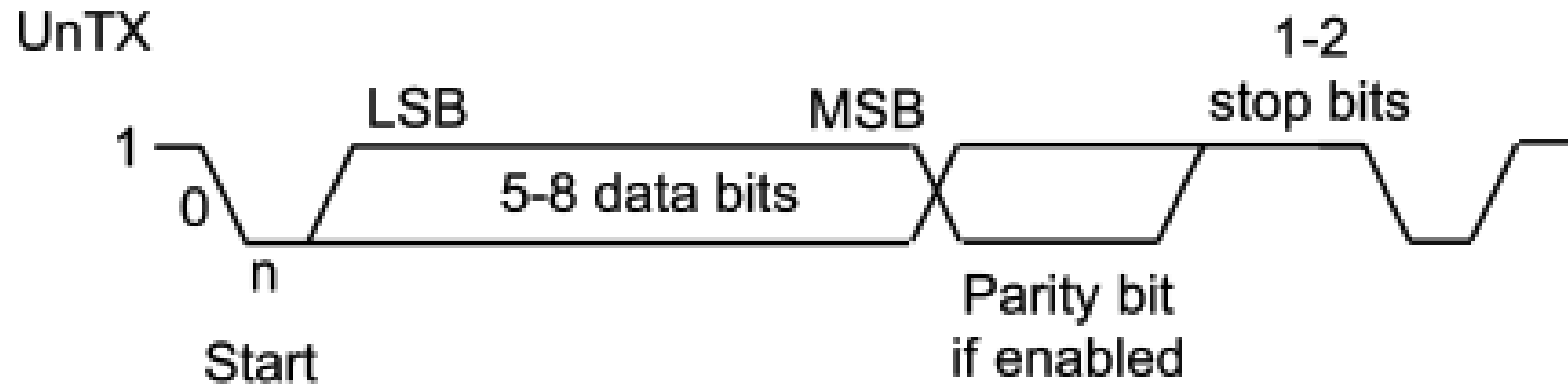
MICRO-
CONTRO-
LADORES
ARM

UART



TRAMA DE DATOS

- 5, 6, 7 u 8 bits de datos
- Generación / detección de bits pares, impares, fijos o sin paridad
- Generación de 1 o 2 bits de parada



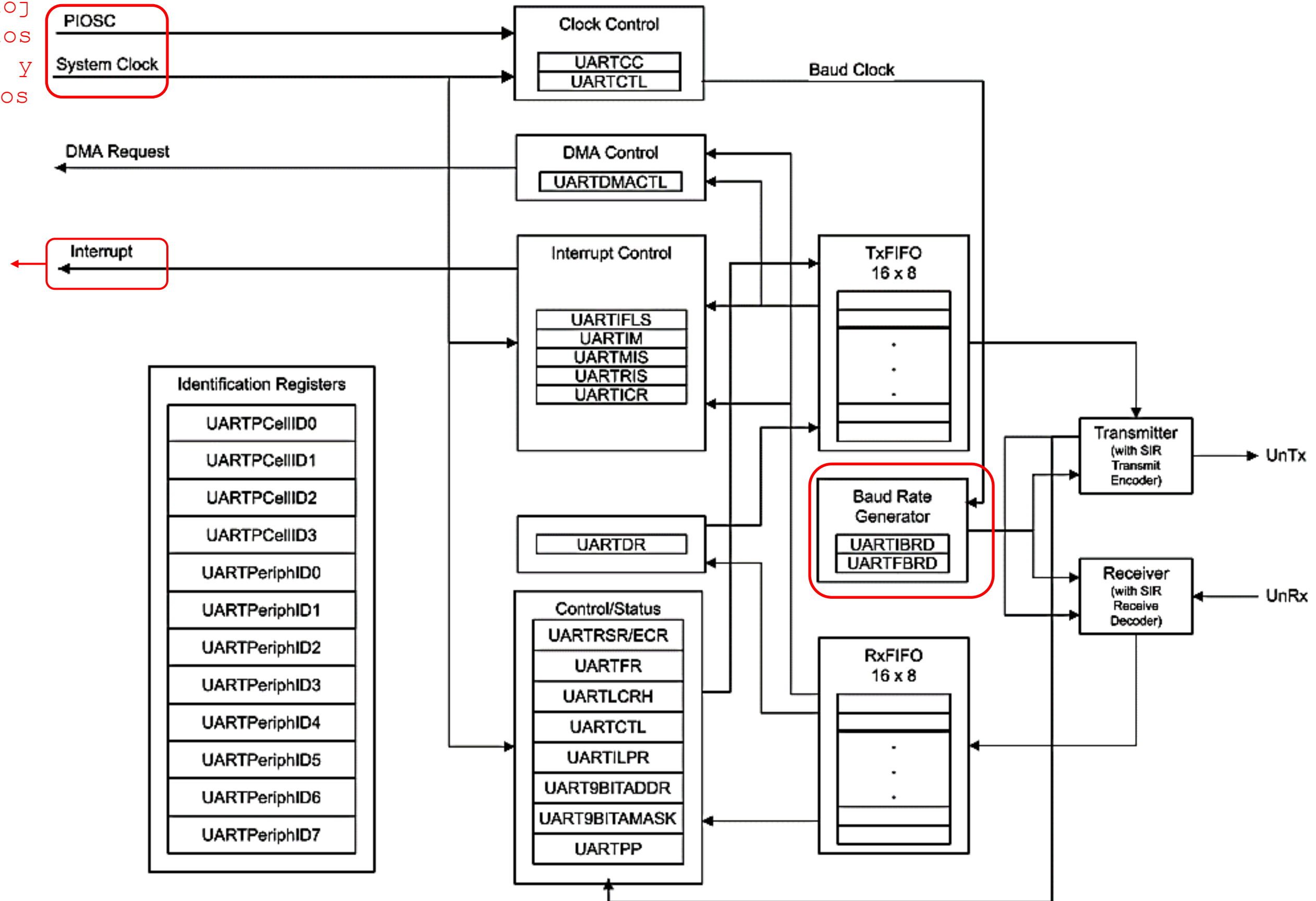
arm

MICRO-
CONTRO-
LADORES
ARM

UART

Fuente de reloj
para acceso a los
registros y
generación baudios

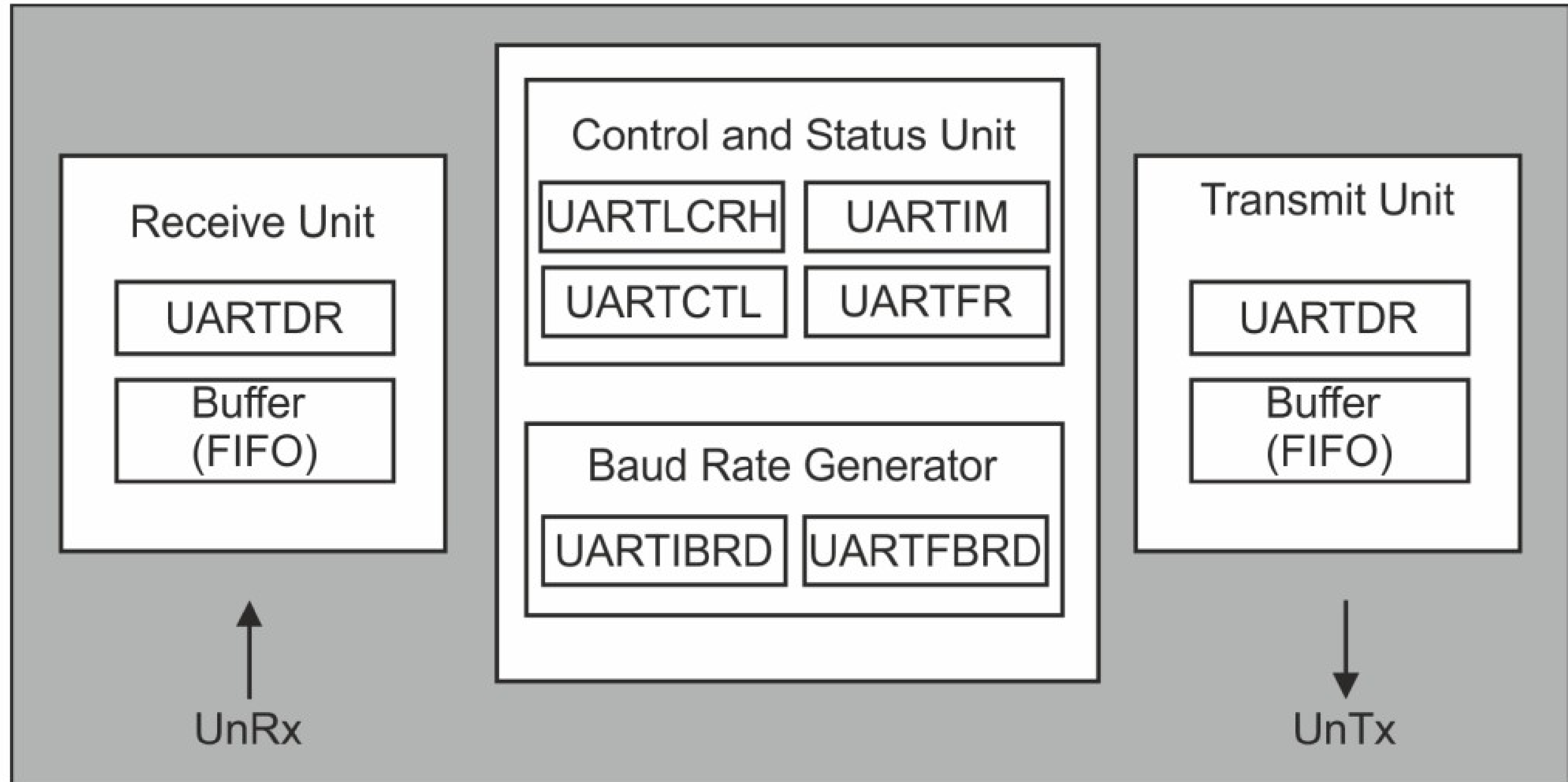
Generación de
interrupciones



arm

MICRO-
CONTRO-
LADORES
ARM

UART



arm

MICRO-
CONTRO-
LADORES
ARM

UART

GENERACION DE BAUDIOS

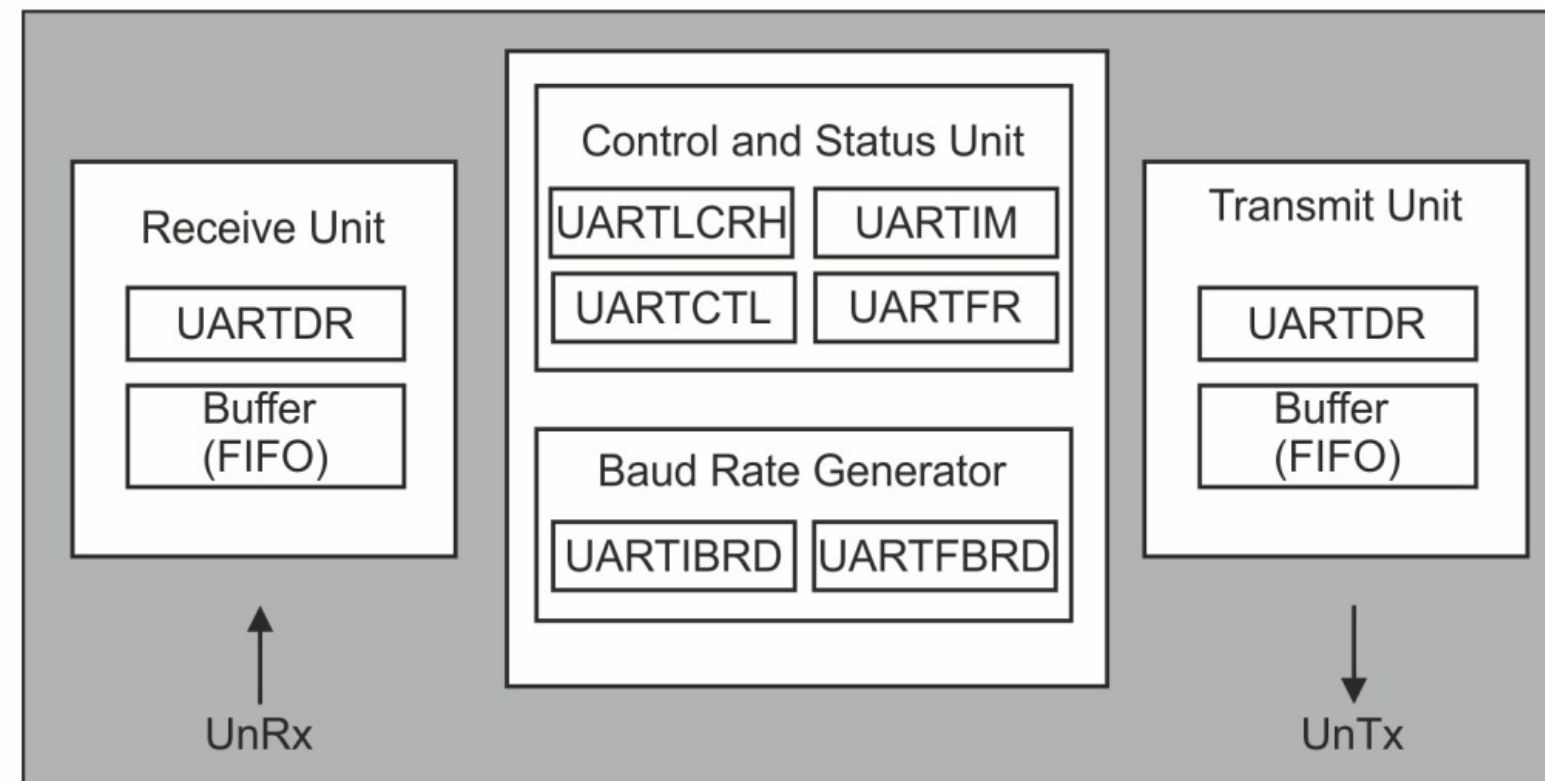
- El divisor de velocidad en baudios es un número de 22 bits que consta de un entero de 16 bits y una parte fraccionaria de 6 bits.
- El entero de 16 bits se carga a través del registro **UART Integer Baud-Rate Divisor (UARTIBRD)** y la parte fraccionaria de 6 bits se carga con el registro **UART Fractional Baud-Rate Divisor (UARTFBRD)**.

$$BRD = BRDI + BRDF = \frac{UARTSysClk}{(ClkDiv * Baud Rate)}$$

$$ClkDiv = 16 \quad \text{si HSE}=0$$

$$ClkDiv = 8 \quad \text{si HSE}=1 \quad (UARTCTL)$$

$$UARTFBRD[DIVFRAC] = \text{integer}(BRDF * 64 + 0.5)$$



arm

MICRO-
CONTRO-
LADORES
ARM

UART

Pin Name	Pin Number	Pin Mux / Pin Assignment	Pin Type	Buffer Type ^a	Description
U0Rx	17	PA0 (1)	I	TTL	UART module 0 receive.
U0Tx	18	PA1 (1)	O	TTL	UART module 0 transmit.
U1CTS	15 29	PC5 (8) PF1 (1)	I	TTL	UART module 1 Clear To Send modem flow control input signal.
U1RTS	16 28	PC4 (8) PF0 (1)	O	TTL	UART module 1 Request to Send modem flow control output line.
U1Rx	16 45	PC4 (2) PB0 (1)	I	TTL	UART module 1 receive.
U1Tx	15 46	PC5 (2) PB1 (1)	O	TTL	UART module 1 transmit.
U2Rx	53	PD6 (1)	I	TTL	UART module 2 receive.
U2Tx	10	PD7 (1)	O	TTL	UART module 2 transmit.
U3Rx	14	PC6 (1)	I	TTL	UART module 3 receive.
U3Tx	13	PC7 (1)	O	TTL	UART module 3 transmit.
U4Rx	16	PC4 (1)	I	TTL	UART module 4 receive.
U4Tx	15	PC5 (1)	O	TTL	UART module 4 transmit.
U5Rx	59	PE4 (1)	I	TTL	UART module 5 receive.
U5Tx	60	PE5 (1)	O	TTL	UART module 5 transmit.
U6Rx	43	PD4 (1)	I	TTL	UART module 6 receive.
U6Tx	44	PD5 (1)	O	TTL	UART module 6 transmit.
U7Rx	9	PE0 (1)	I	TTL	UART module 7 receive.
U7Tx	8	PE1 (1)	O	TTL	UART module 7 transmit.

arm

MICRO-
CONTRO-
LADORES
ARM

ENTRO DE CAPACITACIÓN
E DESARROLLO TECNOLÓGICO

COMUNICACIÓN SERIAL - UART

CONFIGURACION

```
12 #define __GET_IBRD(UARTSysClk, ClkDiv, BaudRate) ((uint16_t)(UARTSysClk/(ClkDiv*BaudRate)))
13 #define __GET_FRACPART(UARTSysClk, ClkDiv, BaudRate) ((float)UARTSysClk/(ClkDiv*BaudRate) - __GET_IBRD(UARTSysClk, ClkDiv, BaudRate) )
14 #define __GET_FBRD(UARTSysClk, ClkDiv, BaudRate) ((uint8_t)( __GET_FRACPART(UARTSysClk, ClkDiv, BaudRate) * 64 + 0.5 ))
15
```

- CMSIS

```
53 /*functions definitions*/
54 void UART0_Config(uint32_t uartSysClk, uint32_t baud){
55
56     /*1. enable clock*/
57     SYSCTL->RCGCGPIO |= 1<<0;           //enable clk for GPIOA
58     while(!(SYSCTL->PRGPIO & 1<<0));    //GPIOA is ready
59     SYSCTL->RCGCUART |= 1<<0;           //enable clk for UART0
60     while(!(SYSCTL->PRUART & 1<<0));    //UART0 is ready
61     /*2. Tx and Rx pinouts configs*/
62     GPIOA->AFSEL |= 1<<1 | 1<<0;        //PA0 and PA1 -> Alternative function
63     /*3. Configure the GPIO current level*/
64
65     /*4. configure the PMCN field in the GPIOPCTL register to assign the uart*/
66     GPIOA->PCTL |= 0x1<<4 | 0x1<<0;     //PA0->RX PA1->TX
67     GPIOA->DEN = 1<<1 | 1<<0;           //PA0, PA1 digital pins
68     /*5. Disable UART0*/
69     UART0->CTL &= ~ (1U<<0);
70     /*6. Write the integer portion of the BRD to the UARTIBRD register.*/
71     UART0->CTL &= ~ 1U<<5;
72     UART0->IBRD = __GET_IBRD(uartSysClk,16,baud);
73     ibrd = __GET_IBRD(uartSysClk,16,baud);
74     /*7. Write the fractional portion of the BRD to the UARTFBRD register*/
75     UART0->FBRD = __GET_FBRD(uartSysClk,16,baud);
76     fbrd = __GET_FBRD(uartSysClk,16,baud);
77     /*8. Write the desired serial parameters to the UARTLCRH register */
78     UART0->LCRH |= 0x3 <<5;             // 8bit data
79
80     /*9. Configure the UART clock source by writing to the UARTCC register.*/
81     UART0->CC &= ~ 0xF;
82     /*10. Optionally, configure the uDMA channel */
83
84     /*11. enable the UART*/
85     UART0->CTL |= 1<<0;
86
87 }
```

- TivaWare

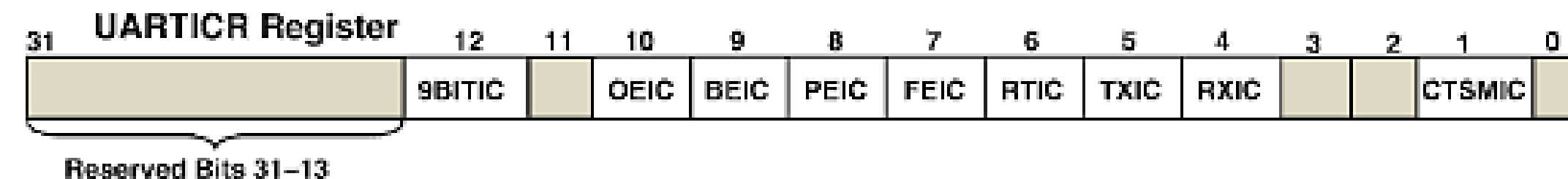
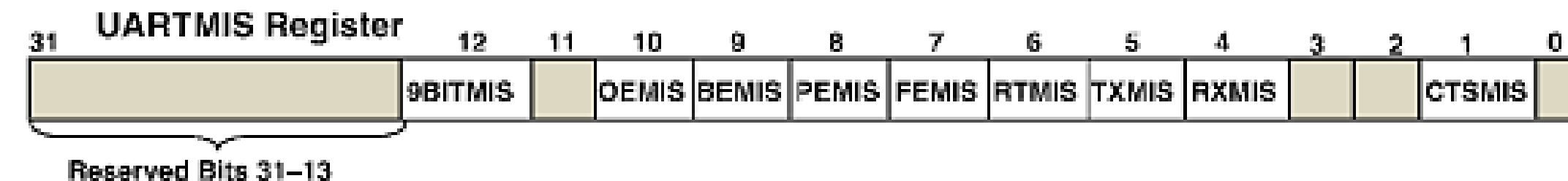
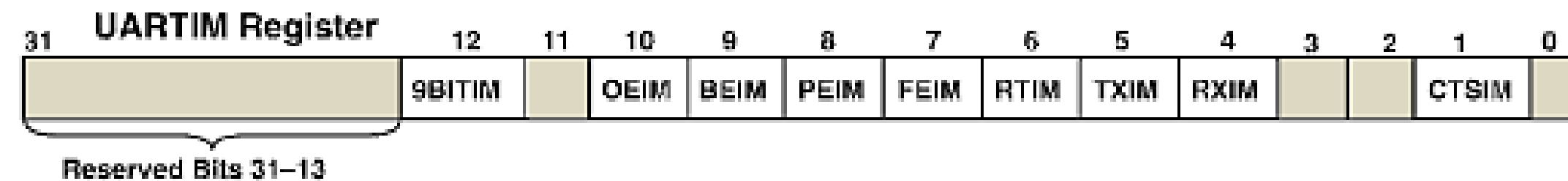
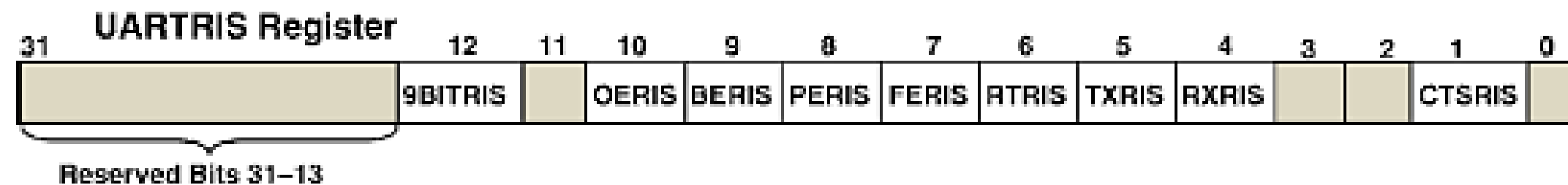
```
80 void UART0_Config(uint32_t uartSysClk, uint32_t baud){
81
82     /*CONFIGURAR LOS PINES RELACIONADOS CON EL UART0*/
83     /*Habilitar el reloj*/
84     SYSCTL_RCGCGPIO_R |= SYSCTL_RCGCGPIO_R0;
85     while(!(SYSCTL_PRGPIO_R & SYSCTL_PRGPIO_R0)); //GPIOA Listo
86     /*2. Establecer las funciones alternativas de los pines respectivos*/
87     GPIO_PORTA_AFSEL_R |= 1<<1 | 1<<0;           //PA0, PA1 -> funcion alternativa
88     /*3. Asignar los pines al UARTx con el registro PCTL*/
89     GPIO_PORTA_PCTL_R |= GPIO_PCTL_PA0_U0RX | GPIO_PCTL_PA1_U0TX;
90
91     /*4. Establecer los pines seleccionados como digitales*/
92     GPIO_PORTA_DEN_R |= 1<<1 | 1<<0;
93
94     /*5. adicionalmente configurar el nivel de corriente de los pines seleccionados*/
95
96     /*CONFIGURAR LA FUENTE DE RELOJ Y EL BAUD RATE DEL UART*/
97
98     /*1. habilitar el reloj del uart0*/
99     SYSCTL_RCGCUART_R |= SYSCTL_RCGCUART_R0;
100    while(!(SYSCTL_PRUART_R & SYSCTL_PRUART_R0));
101    /*2. deshabilitar el uart0*/
102    UART0_CTL_R &= ~ UART_CTL_UARTEN;
103    /*3. Establecer el valor entero del en le registro IBRD*/
104    UART0_IBRD_R = __GET_IBRD(uartSysClk, 16, baud);
105    /*4. establecer la pare fraccionaria en el registro FBRD*/
106    UART0_FBRD_R = __GET_FBRD(uartSysClk, 16, baud);
107
108    /*5. escribir los parametros deseados de la comunicacion serial*/
109    UART0_LCRH_R |= UART_LCRH_WLEN_8;
110    /*6. seleccionar la fuente de reloj para el UART0*/
111    UART0_CC_R &= UART_CC_CS_SYSClk;
112
113    /*7. habilitar el UART0*/
114    UART0_CTL_R |= UART_CTL_UARTEN;
115 }
```

COMUNICACIÓN SERIAL - UART

INTERRUPCIONES

Los registros para el manejo de interrupciones son:

- UART Interrupt FIFO Level Select (**UARTIFLS**) Register.
- UART Raw Interrupt Status (**UARTRIS**) Register.
- UART Interrupt Mask (**UARTIM**) Register.
- UART Masked Interrupt Status (**UARTMIS**) Register.
- UART Interrupt Clear Register (**UARTICR**).
- UART DMA Control (**UARTDMCTL**) Register.

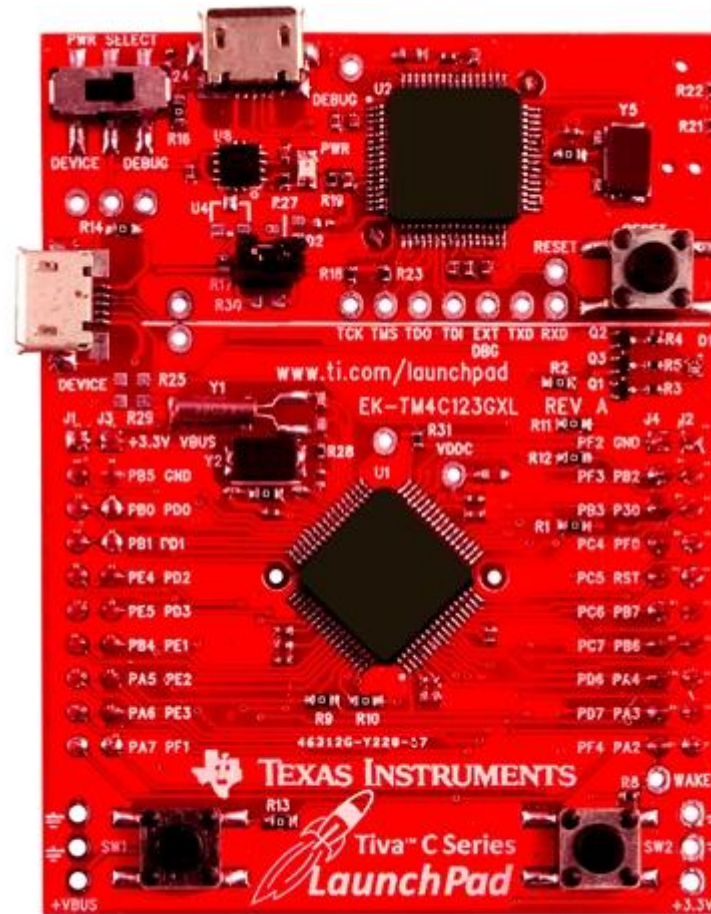


arm

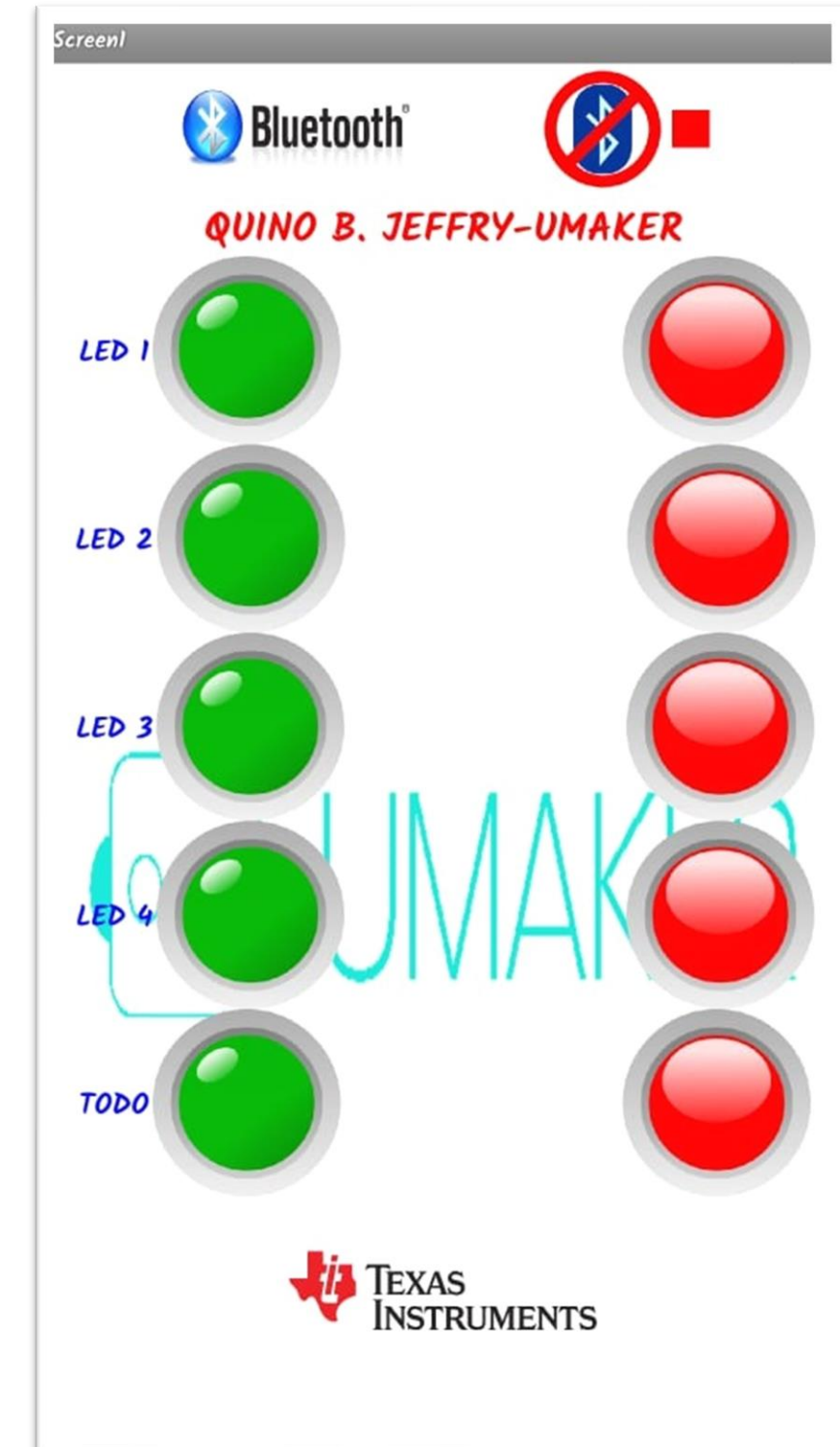
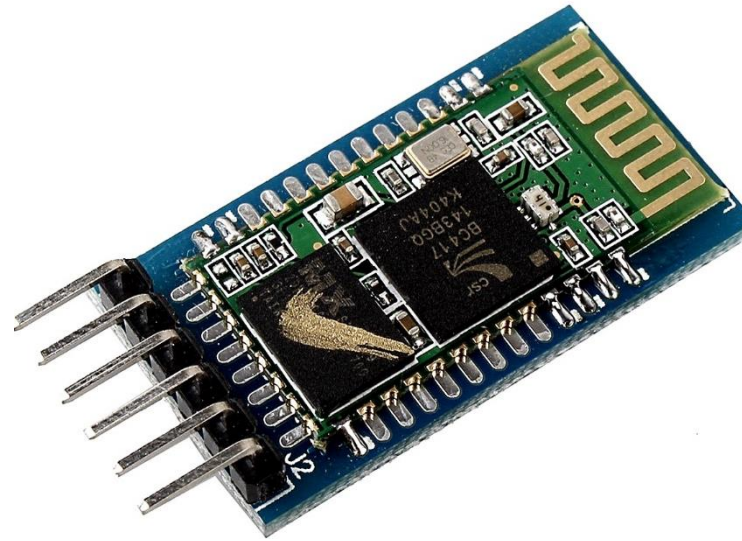
MICRO-
CONTRO-
LADORES
ARM

COMUNICACIÓN SERIAL - UART

Aplicativo Android y comunicación con el modulo bluetooth HC05



UART3
U3Tx → PC7
U3Rx → PC6



arm

MICRO-
CONTRO-
LADORES
ARM

UMAKER | CENTRO DE CAPACITACIÓN
DE DESARROLLO TECNOLÓGICO